

REMARKS

This Amendment and Response is intended to fully respond to the Final Office Action dated May 5, 2004. In that Office Action, claims 1-6 and 8-28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,995,756 to Herrmann in view of U.S. Patent No. 6,594,682 to Peterson et al.

Reconsideration of the rejections, in view of these remarks, is respectfully requested.

Brief Description of the Invention

Before addressing the claim rejections based on the prior art, a brief description of the invention is hereby provided. The present invention relates generally to methods and systems for identifying files that a user of a client computer may need when the client computer is disconnected from a network. The files may be automatically or manually identified and may be related to file usage patterns. In order to execute the identified files on the client computer when it is disconnected from the network, before the disconnection from the network occurs, a handler associated with each file type identifies application software that may be required to execute each of the files. Each of the files (if not on the client computer already) and their associated application software is downloaded to the client computer. This is particularly helpful when most if not all the functionality typically occurs on a remote computer system, such as through the use of remote procedure calls, and very little of the application functionality has been installed on the client computer system. Such use of other network or remote computers to perform application functionality is an increasing phenomenon and therefore the problems associated with disconnecting a client computer system from the network have increased significance and hence, the motivation for the present invention.

Another particular problem that is solved by the present invention relates to the fact that some applications download or install functionality to a client computer “on demand.” That is, when connected to a network, a particular user may be working with a file along with some of its associated application, wherein that portion of the *application functionality* has been downloaded to the computer system. However, some of the other application functionality of that application has not been downloaded to the client computer system since the user, in this example, has not

requested the other functional component(s). Upon requesting the functionality, then and only then would that functionality be downloaded/transferred to the client computer system and installed, i.e., the functionality is installed on demand. (See Specification, p. 8, l. 34 through p. 9, l. 3). In this example, if the user disconnects from the network before requesting such functionality, then that functionality will not be available while the computer is disconnected from the network. The present invention solves this problem by identifying possible functionality that may be needed while working with the file offline and transfers such functionality to the client computer system before the computer is disconnected from the network. Of course, other problems are also solved using the present invention, but this particular problem highlights the nature of the invention.

Claim Rejections - 35 U.S.C. § 103

Claims 1-6 and 8-28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,995,756 to Herrmann in view of U.S. Patent No. 6,594,682 to Peterson et al. Applicant respectfully traverses the §103 rejection as the Examiner has failed establish a prima facie case of obviousness. Indeed, such a prima facie case can only be met when **all** of the following requirements are met: (1) there must be some suggestion or motivation in the references themselves (or in the knowledge available to those skilled in the art) to combine the references; (2) there must be a reasonable expectation of success; and (3) the combined references must teach or suggest all the claim limitations. See MPEP §§ 706.02(j) and 2143. The combination of Herrmann and Peterson et al. does not teach or suggest all the claim limitations, there is no suggestion or motivation to combine the references, and there is no reasonable expectation of success.

REBUTTAL OF PRIMA FACIE CASE OF OBVIOUSNESS

Applicant respectfully traverses the §103 rejections. The Examiner has failed to substantiate a prima facie case of obviousness because one of the requirements of a prima facie case is absent. In particular, the cited references do not teach or suggest all the limitations of

independent claims 1, 4, 10 or 21. Specifically, neither Herrmann or Peterson, alone or in combination, disclose, teach or suggest the limitation related identifying application functionality needed to run when the computer is disconnected from the network and, in the case of claims 4, 10 and 21 the limitation of transferring and/or installing application functionality to the client computer system not previously transferred so that the application functionality is available when the client computer system is disconnected from the network.

Specifically, Herrmann describes a system and method for executing an application located on a remote computer system, wherein the application has been divided into frames, and wherein each frame has its own functionality. (See Col. 7, ll. 49-60 of Herrmann) As best understood, the user essentially operates his/her browser to access specific frames of the application. When a new frame is requested, an HTTP request is invoked to get the proper functionality, which comes in the form of another MIME file. The MIME file includes information as to how to download the operational code, and such downloading occurs shortly after receiving the MIME file. Herrmann states “When the user selects a hyperlink that points to an application page, the Web browser downloads the application code and executes the page inside the browser, as FIG. 4C illustrates.” (Col. 9, ll. 38-41) The execution involves the browser invoking a handler to check locally for the application code and then, if not found locally, downloading the code. However, not all components are downloaded at the same time. Herrmann states:

“When the code is downloaded, a ‘code base’ specification (file) is initially requested from the server. The code base itself can range from a simple DLL file to a Cabinet file (Microsoft .cab file) containing multiple compressed files. Still further, an information (e.g., Microsoft .inf) file can be employed for instructing the client system how to install the downloaded application. These mechanisms afford great flexibility in choosing which component of an application gets downloaded and when.” (Herrmann, Col. 9, ll. 57-65, emphasis added here).

As a result, Herrmann provides a system of accessing code on an “on-demand” basis. This is similar to the on-demand application described in the specification of the present application wherein only portions of the functionality are downloaded at a time.

Contrary to the pending claims however, Herrmann does not teach or suggest a system or method that identifies application functionality to be used when the computer is disconnected from the network, independent from previously requested functionality. Herrmann specifically

does not relate to identifying functionality based on an analysis of files and the potential functionality associated with or related to the files, let alone transferring and installing such functionality to and on the client computer system for use when the computer is disconnected from the network.

Peterson does not make up for the inadequacies of Herrmann. Peterson provides a system wherein a user can request web content while offline, the web content is then retrieved when the user reestablishes a connection, i.e., the cached requests are satisfied. In order to achieve the downloading of web content Peterson requires a scheduling subsystem. This client-based module is used to establish a connection with a server based on a predetermined schedule (albeit possibly with some level of randomness) to request web content. The scheduling subsystem does not analyze an existing file or determine whether functionality is needed while the user is going to be offline. In fact the entire system of Peterson is centered around the user requesting specific web content, whether through indexes, filter preferences, etc., and does not provide a handler associated with each file type to identify application software that may be required to execute each of the files.

Peterson does provide for the browsing of the retrieved web content while the user is offline. However, there is no indication that any specific functionality associated with browsing the web content is provided to the user to allow such browsing to occur. In essence, Peterson only provides one small portion of a similar problem, i.e., having the requested content offline. That said however, the claims of the present invention deal much more specifically with the ability to identify, generally automatically, the needed functionality to execute items. Since Peterson does not identify any specific content, based on foreseeable needs, to execute a particular file, it does not teach or disclose such an element.

Thus, the combination of Herrmann and Peterson simply does not teach or suggest each of the elements of the claimed invention. Neither Herrmann and Peterson, alone or in combination, the limitation related identifying application functionality needed to run when the computer is disconnected from the network and, in the case of claims 4, 10 and 21 the limitation of transferring and/or installing application functionality to the client computer system not previously transferred so that the application functionality is available when the client computer system is disconnected from the network. Given that these references, either alone or in

combination, fail to disclose, teach or suggest all the claim limitations, they cannot, as a matter of law, render the independent claims obvious. Reconsideration of the § 103(a) rejections is therefore respectfully requested.

Moreover, Applicant traverses the §103 rejections since there is no suggestion within the references themselves to combine the references. That is, Herrmann does not even consider the situation where the client becomes disconnected from the network. Based on the teaching of Herrmann, only code that is to be used in accordance with a specific request should be downloaded. Indeed, Herrmann states: “Once the application is executing at the client, such as illustrated in FIG. 4D, it can execute remote logic such as using RPC (Remote Procedure Call) methodology. In this manner business logic which is preferably implemented as remote procedures can still be used, as before.” (Herrmann, Col. 10, ll. 20-25). Clearly, Herrmann does not contemplate transferring all the functionality to the client computer system and most importantly, Herrmann does not contemplate transferring functionality such that the client computer system can operate offline. Moreover, Herrmann does not suggest the desirability of the present invention. (See MPEP 2143.01 “The prior art must suggest the desirability of the claimed invention.”) Given this, there is simply no suggestion within Herrmann to combine its teachings with those of another reference concerned with offline browsing of content, e.g., Peterson.

If anything, Herrmann teaches away from the present invention in that it relies on user connectivity with the network to provide application information while maintaining the remote procedure call, or “business logic” methodology in place. Moreover, Herrmann further states:

“Once the form is created, it can establish connections back to any remote server objects it needs to perform its functions. At this point, the user can interact with the form, which will appear embedded in the Internet Explorer frame. When the user changes to a different page, the browser assume [sic] responsibility for eventually closing and destroying the form (and relinquishing any outstanding connection to the remote servers.) (Herrmann, col. 9, ll. 4-11).

It is clear that Herrmann only contemplates the use of any downloaded functionality as part of an “online” session and thus does not suggest the use of such functionality offline. Herrmann does not suggest the analysis of a file to determine if some functionality should be downloaded such that the functionality is available once the computer is disconnected from the

network. Moreover, there is no suggestion within Peterson to suggest combining its teachings with Herrmann. The Examiner stated that the motivation to combine the references “is to allow a user [to] browse through Web content while offline, in the same manner that he/she browses the content while online (Col. 12 Ln. 41-43)” (Office Action, p. 11). Although Peterson describes this aspect as part of *Peterson’s* invention, this does not necessarily relate to the motivation to combine it with a system that does not contemplate offline processing. Moreover, there is no suggestion that one skilled in the art would combine a reference concerned with caching web content with a document concerned with online processing functionality. Thus, the Examiner has failed to point to any teaching or suggestion in Herrmann and Peterson that would teach or suggest the analysis of a file to then download functionality such that the functionality would be available offline, i.e., for the purpose of operating offline.

Additionally, there can be no expectation of success when combining the two systems of Herrmann and Peterson. Indeed, assuming *arguendo* that the references were combined, a best case combination would result in a system that requires a client-side scheduling system to manage requests (specifically made by the user) for functionality, the scheduling system requesting such functionality during the next server communication, i.e., the next time the user is online. That is, since neither Herrmann nor Peterson identify required functionality to download of offline use as part of their specific inventions, combining the two systems would not result in a system that operates like the presently defined invention.

For the above reasons, Applicant respectfully requests that the § 103(a) rejections be withdrawn and that the pending claims 1-6 and 8-28 be allowed.

Conclusion

It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

In light of the above remarks and amendments, it is believed that the application is now in condition for allowance, and such action is respectfully requested. Should the Examiner have

any remaining questions or concerns, he is encouraged to contact the undersigned attorney by telephone to expeditiously resolve such concerns.

Respectfully submitted,

Dated: _____

6/29/04



T. B. Scull
Timothy B. Scull, Attorney Reg. No. 42,137
MERCHANT & GOULD P.C.
P. O. Box 2903
Minneapolis, MN 55402-0903
(303) 357-1648